

FACTA UNIVERSITATIS

Series: **Automatic Control and Robotics** Vol. 19, N° 3, 2020, pp. 175 - 190

<https://doi.org/10.22190/FUACR2003175P>

THE APPROACH TO MEASUREMENT OF REQUIREMENT QUALITY BY APPLICATION OF GENERALIZED PRIORITIZED FUZZY CONSTRAINT SATISFACTION PROBLEM

UDC (681.5.01:004.4)

Radomir Prodanović¹, Dejan Rančić², Ivan Vulić³, Dušan Bogičević³

¹Center for Applied Mathematics and Electronic, Serbian Armed Forces, Belgrade,
Republic of Serbia

²University of Niš, Faculty of Electronic Engineering, Niš, Republic of Serbia

³University of Defence, Military Academy, Belgrade, Republic of Serbia

Abstract. *The requirement quality affects product development at all lifecycle stages, as well as the end product. Poorly defined requirements bring to extended deadlines, increased financial costs, even to project disruption. Current researches related to the good quality of requirements include characteristics of good requirements and the development of new elicitation techniques. Requirement quality evaluation should be tailored both to the professionals and users who defined requirements according to their needs. Therefore, the model is designed for requirement quality measurement based on the characteristics of good requirements by application of the Generalized Prioritized Fuzzy Constraint Satisfaction Problem. The model enables the participation of selected characteristics of good requirements in quality evaluation, according to priorities. The evaluator obtains information if the requirement satisfies the given quality satisfaction threshold based on the degree of fulfillment of selected characteristics of a good requirement. The model is applied to all types of requirements, as well as to the evaluation of requirements at all software development lifecycle stages.*

Key words: *Quality requirement, GFPCSP, fuzzy logic, elicitation requirement, software product*

1. INTRODUCTION

A complex software product brings a bigger risk of failure in all phases of its product development life cycle. The issue with the design of such a complex product is the identification of all requirements and their elaboration with regard to the quality. According to

Received November 06, 2020

Corresponding author: Radomir Prodanović

Center for Applied Mathematics and Electronic, Serbian Armed Forces, Vojvode Stepe 445, 11000 Belgrade,
Republic of Serbia

E-mail: radomir.prodanovic@vs.rs

the Chaos Report [1], 29% of projects completes successfully, 52% completes with issues while 19% completes unsuccessfully. The Report points out that incomplete requirements are in most cases the key factor for the project unsuccessful completion. Earlier research of the Standish Group from 1994 and 1995 also confirms that incomplete requirements are the cause of the project failure. It is obvious that the issue of the quality of requirements has been longstanding now. At the same time, reports show that clearly defined requirements are one of the key factors contributing to successful project completion.

About improved elicitation of requirements, numerous techniques that overcome difficulties of elicitation, and definition of the good quality requirements have been designed [2-7]. Classification schemes of requirements for the software product are among the most significant. The classification scheme of software requirements [8] provides a working framework in which software requirements are elicited and defined.

The appropriate quality of a requirement should be achieved by the application of the techniques for elicitation of requirements and classification requirement schemes. The quality of a requirement implies the realization of the “characteristics of good requirements”. The good quality requirement affects positively the development of software at all stages of a product’s lifecycle and the end quality of the product [9]. There are several classifications of a good quality requirement characteristics described in the literature, but the most significant are completeness, clarity, and verifiability.

There is an issue of how to determine if a requirement is of good quality for further stages of software development. Measuring the quality of the requirement helps in the reduction of risks stemming from poorly defined requirements that are ambiguous and may cause an error in product development. Well-defined requirements directly influence the quality of the software product and implementation timeframe.

The paper [10] identifies indicators for measuring based on the characteristics of good quality requirements. B. Heinrich et al. [11] propose a set of five requirements for data quality metrics to support both decision-making under uncertainty and an economically oriented management of data quality. Robert J. Halligan [12] presents a structured methodology for measuring the quality of requirements, individually and collectively. Since the majority of software requirements today are still written in natural language, these approaches focus on measurable indicators that can be derived from the text. The paper [13] particularly addresses the relationships between textual indicators and individual quality attributes. Wang et al. [14] establish relations between activities of the elicitation and quality for good requirements so that the quality is being evaluated in each activity of the elicitation process.

The issue here is how to successfully evaluate the good quality requirements, that is if the requirement has been written and defined in a way that both users and engineers understand it. To solve the issue, the authors apply fuzzy logic for estimation of the required quality evaluating the quality of each selected characteristic of a requirement. The purpose of this research is to design a model for evaluation of the quality of requirements that would enable evaluators and users of the requirement a fast and easy quality estimation.

The remaining of the paper is organized according to the following structure: the second section briefly describes characteristics of good requirements; the third section describes the Generalized Prioritized Fuzzy Constraint Satisfaction Problem (GFPCSP) while the fourth section describes the model for quality measurement of requirement with the application of GFPCSP. The practical application of the model in the quality measurement of a requirement with six characteristics of good requirements is described in the fifth section, while chapter six provides a conclusion.

2. CHARACTERISTIC OF GOOD REQUIREMENTS

The term requirement has been in use in the software engineering community since at least the 1960s [15]. According to the Guide to the Business Analysis Body of Knowledge® version 2 from IIBA (BABOK), [16] a requirement is:

1. A condition or capability needed by a stakeholder to solve a problem or achieve an objective.
2. A condition or capability that must be met or possessed by a solution or solution component to fulfill a contract, standard, specification, or other formally imposed documents.
3. A documented representation of a condition or capability as in (1) or (2).

This definition is based on IEEE 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology [17].

The characteristics of good requirements are variously stated by different writers, with each writer generally emphasizing the characteristics most appropriate to their general discussion or the specific technology domain being addressed. However, the following characteristics are generally acknowledged [18,19]. Characteristic are: unitary or cohesive, complete, consistent, non-conjugated or atomic, traceable, current, unambiguous, specific importance, verifiable.

The paper [14] provides an overview of good characteristics that affect the quality of requirements, as follows: complete, consistent, correct, unambiguous, verifiable, gradable, modifiable, traceable, understandable, feasible, clear, independent, non-redundant, brief (terse), implementation free, necessary.

The list may be extended to include the following favorable characteristics: accurate, priority, unique [20], cohesiveness, currency, external observability, mandatory, relevance, usability, validity [21].

3. GENERALIZED PRIORITIZED FUZZY CONSTRAINT SATISFACTION PROBLEM

Generalized Prioritized Fuzzy Constraint Satisfaction Problem is an expansion of the Prioritized Fuzzy Constraint Satisfaction Problem (PFCSP)[22] which relates to the application of disjunction and negation. PFCSP systems enable the application of conjunction, that is t-norm on constraints. To expand PFCSP to GPFCS disjunction implies t-conorm that is dual to t-norm selected for conjunction, while the negation implies the standard negation: $N(x) = 1-x$. The expansion is realized by the expansion of the third and fourth PFCSP axioms while there is no need to expand other axioms as they relate to the conjunction. Changes in the third axiom relate to the addition of disjunction, while the fourth axiom is generalized so that the monotony is valid only when negation is not being used. Extended definition and its axioms are provided down below and are taken from [23].

Definition 1. X, D, C^f, ρ, g is defined as:

1. Set $X = \{x_i \mid i = 1, 2, \dots, n\}$ is final set of variables.
2. Set $D = \{D_i \mid i = 1, 2, \dots, n\}$ is final set of the range. Each range D is a set of elements that may take variables x from the set X .
3. Set of fuzzy constraints C^f is presented by fuzzy relations whose functions are defined as follows:

$$\mu_{R_i^f}: \left(\prod_{x_j \in \text{var}(R_i^f)} D_j \right) \rightarrow [0, 1],$$

whereas $\text{var}(R_i^f)$ presents the set of variables in the stated constraints R_i^f .

4. ρ is priority function $\rho: C^f \rightarrow [0, \infty)$.
5. Function g aggregates priorities of each constraint with the value of the constraint. Values aggregate in this way are being aggregated in the global degree of constraint satisfaction by aggregation operator \oplus , while GPFCSPP represents the following nine $(X, D, C^f, \rho, g, \wedge, \vee, \neg)$.

Elementary formula is the ordered pair $(x, \rho(R_i^f))$ with $R_i^f \in C^f$, and $x \in \text{Dom}(R_i^f)$

is the degree of the constraint satisfaction for R_i^f , while $\rho(R_i^f)$ is its priority.

The formula GPFCSPP-a is defined as follows:

1. The elementary formula is a formula.
2. If f_1 and f_2 are formulas, then $\wedge(f_1, f_2)$, $\vee(f_1, f_2)$ and $\neg(f_1)$ are also formulas.

The degree of constraint satisfaction $\alpha_F(\vartheta_x)$ for a valuation ϑ_x is calculated concerning the interpretation of the conjunction.

The system is GPFCSPP if it is:

1. If it is $F = \wedge_{i \in \{1, \dots, n\}} f_i$ GPFCSPP formula, with $f_i, i \in \{1, \dots, n\}$ being elementary formulas and C^F is a set of constraints present in the formula, and if for fuzzy constraint R_{max}^F is $\rho_{max} = \rho_{max}(R_{max}^F) = \max\{\rho(R^F | R^F \in C^F)\}$, then it is, also, for every F formula $\mu_{R_{max}^F}(\vartheta_{\text{var}(R_{max}^F)}) = 0$ and then $\alpha_F(\vartheta_x) = 0$.
2. If it is $\exists \rho_0 \in [0, 1]$ so that $\forall R^f \in C^f$ it is also $\rho(R^f) = \rho_0$, then the global degree of constraint satisfaction is as: $\alpha_F(\vartheta_x) = F_\wedge(\vartheta_x)$, whereas F_\wedge is an interpretation of a logic formula F in the fuzzy logic.
3. If we assume that for $R_i^f, R_j^f \in C^f$ it is $\rho(R_i^f) \geq \rho(R_j^f)$, if it is $\delta > 0$ and there are two combined valuation ϑ_x i ϑ'_x , so that for $\forall R^f \in C^f$ it is:
 - a. if $R^f \neq R_i^f$ and $R^f \neq R_j^f$, then $\mu_{R^f}(\vartheta_{\text{var}(R^f)}) = \mu_{R^f}(\vartheta'_{\text{var}(R^f)})$
 - b. if $R^f = R_i^f$, then $\mu_{R^f}(\vartheta_{\text{var}(R^f)}) = \mu_{R^f}(\vartheta'_{\text{var}(R^f)}) + \delta$
 - c. if $R^f = R_j^f$, then $\mu_{R^f}(\vartheta_{\text{var}(R^f)}) = \mu_{R^f}(\vartheta'_{\text{var}(R^f)}) + \delta$

Then, if it is:

$$F = \wedge_{k=1, \dots, n} (x_k, \rho(R_k^f)), \quad x_k \in \text{Dom}(R_k)$$

$$F = \vee_{k=1, \dots, n} (x_k, \rho(R_k^f)), \quad x_k \in \text{Dom}(R_k)$$

It follows:

$$\alpha_F(\vartheta_x) \geq \alpha_F(\vartheta'_x).$$

4. If given two different valuation combinations ϑ_x i ϑ'_x with the characteristic that $\forall R^f \in C^f$ is $\mu_{R^f}(\vartheta_{\text{var}(R^f)}) \geq \mu_{R^f}(\vartheta'_{\text{var}(R^f)})$. Then, if the F formula does not include negation, it is $\alpha_F(\vartheta_x) \geq \alpha_F(\vartheta'_x)$.

5. If there is a combined variation ϑ_x such that $\forall R^f \in C^f$ then it is $\mu_{R^f}(\vartheta_{var(R^f)}) = 1$ and if F is a formula $F = \bigwedge_{i \in \{1, \dots, n\}} f_i$ or $F = \bigvee_{i \in \{1, \dots, n\}} f_i$, whereas $f_i, i \in \{1, \dots, n\}$ are elementary formulas, then it is $\alpha_F(\vartheta_x) = 1$.

Takaci in his paper [24] describes and proves one special GPFCS system.

Theorem 2. The following system $(X, D, C^f, \rho, g, \wedge, \vee, \neg, \rightarrow)$ whereas $\wedge = T_L, \vee = S_L, \neg = N_S$ and finally, $\diamond(x_i, p_i) = S_p(x_i, 1 - p_i)$ is GPFCS. Global degree of constraint satisfaction of valuation v_x for F formula is calculated as follows:

$$\alpha_x(\vartheta_x) = \Phi \left\{ \diamond(v_{x_i}) \frac{\rho(R_i^f)}{\rho_{max}} \mid R^f \in C^f \right\} \tag{1}$$

Whereas C^f is a set of the constraint formula $F, \rho_{max} = \max\{\rho(R_i^f), R^f \in C^f\}$, while Φ is an interpretation of the F formula in GPFCS.

4. REQUIREMENT QUALITY MEASUREMENT MODEL WITH APPLICATION OF GPFCS

The model was created as a result of the analysis of the process of requirement elicitation and definition for implementation of PKI infrastructure in the Ministry of Defense and the Serbian Armed Forces, as well as issues aroused from poorly defined requirements at all lifecycle stages of development of the software for PKI. Based on the model, see Fig. 1, requirement quality evaluators or a person defining the requirement may estimate if the requirement meets characteristics of good requirements and thus define the good quality requirement that will not slow down implementation and further advancement of the project.

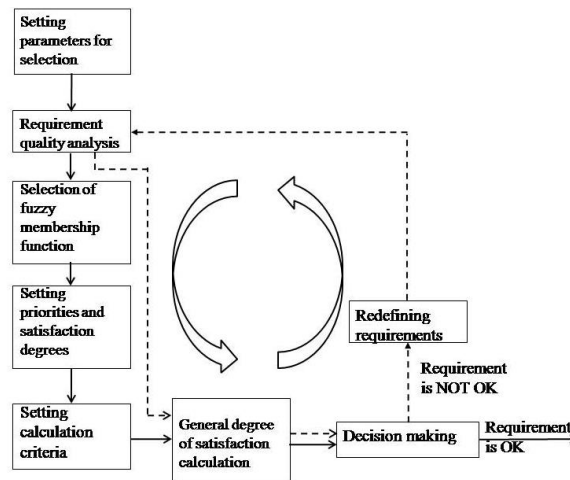


Fig. 1 Quality requirement measurement model based on GPFCS

Identified characteristics of good quality requirements not implemented in the requirement specification, but incomplete or poorly defined, served as the foundation for model definition. Characteristics of good requirements and GPFCS are the key basics of the model.

4.1. Setting the parameters for decision-making on the requirement quality

Parameters for decision-making on the requirement quality are selected from the characteristics of a good requirement. There are several lists of characteristics for the definition of good requirements in literature. Selection of a larger number of characteristics will affect the better judgement of the requirement quality, but will, also, cause increased complexity of the process and measurement of the requirement quality. Selected parameters, meaning characteristics of a good requirement, are linguistic variables in the application of the fuzzy logic.

4.2. Quality analysis of requirements based on the selected parameters

The quality of requirement is determined by evaluating each selected parameter by answering the questions on the estimation of the quality of parameters. There are several different lists of questions for evaluation of satisfaction of characteristics of good quality requirements in literature [21, 25].

Based on the answer, the requirement for the particular parameter may not fulfill, may partially fulfill, fulfill, and partially not fulfill the quality. If the parameter partially fulfills the quality this means that it does not fulfill the quality in a larger part, while fulfilling it in its smaller part. In case that the parameter partially does not fulfill the quality, then it means that it fulfills the quality in its larger part, while does not fulfill it in its smaller part. Partial fulfillment or not fulfillment of the quality of a parameter is expressed in percentage by the evaluator. For example, the parameter does not fulfill but fulfills with 20% the quality, or the parameter fulfills the requirement quality but does not fulfill it with 10%.

A situation like these in measuring the degree of fulfillment of parameters are created when the evaluator cannot precisely determine based on the answers obtained from the question list if the parameter fully satisfies the quality or not.

The evaluator of the requirement quality evaluates the degree of parameter fulfillment for each of the questions. If the evaluator answers that one of the values “fulfills” or “does not fulfill”, then the answer may be one of the partial values. The evaluator then evaluates to what percentage it answers the question. Upon all answers obtained, the evaluator makes the final decision on fulfillment of parameters. For example, the parameter partially fulfills with 30% meaning that it has not been fulfilled in the larger part, but only in the smaller part.

4.3. Definition of the fuzzy membership functions

Fuzzy membership functions are distinguished functions that primarily define a fuzzy set. The objective of the function is to associate the degree of membership to every element from the universe of the corresponding fuzzy set. There are different types of membership functions, like trapezoidal, triangular, Gaussian-like, logical. Membership functions are designed based on the research results. Experts usually choose or design freely membership functions needed to resolve the issue in a way to meet the following conditions:

- The domain of the membership function is in the range $[0,1]$. Membership function domain is never a negative one.
- Injectivity, meaning that each element x from the fuzzy set X function $\mu_A(x)$ transforms in different values. The same element of the fuzzy set cannot have different membership values for the same fuzzy set.

Triangular and trapezoidal membership functions are the most used ones. Fig. 2 and Fig. 3 show triangular and trapezoidal membership functions.

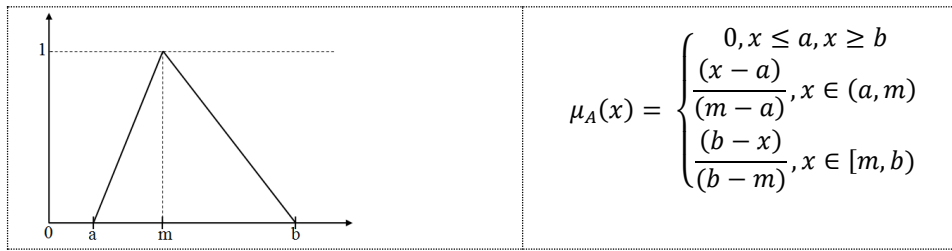


Fig. 2 Triangular membership function

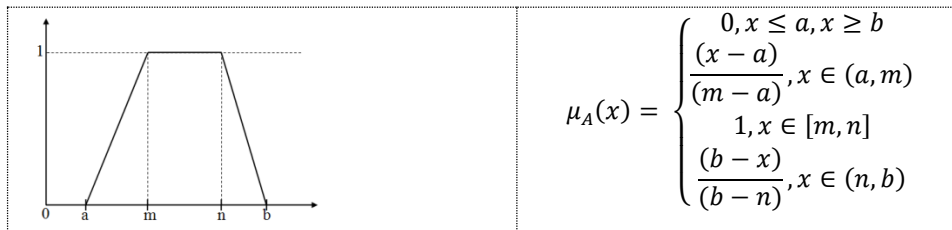


Fig. 3 Trapezoidal membership function

Parameters for evaluation of the requirement quality are linguistic variables when translated into fuzzy logic. To calculate the linguistic variable, it is necessary to determine membership functions for each linguistic value. Evaluators can define their membership functions due to their experience. Table 1. shows that linguistic variables are of the following linguistic values: fulfills, does not fulfill, partially fulfills and partially does not fulfill. Triangular and trapezoidal functions are most suitable for the design of membership functions of linguistic values.

4.4. Calculation of the membership function value for the model

Membership functions for linguistic values may be described by the left (L) and right (R) trapezoidal or triangular interval. Thus, the membership functions for linguistic values “does not fulfill” or “partially fulfills” may be described by an L trapezoidal or a triangular fuzzy interval (a, m), while the membership functions for linguistic values “fulfill” and “partially does not fulfill” may be presented by the R trapezoidal fuzzy interval (0, b), or the R triangular fuzzy interval (m,b), as shown in Fig. 4 and Fig. 5.

In the interval (m, n] for the trapezoidal function and m for the triangular function, the membership function for the linguistic value “fulfills” is $\mu_A(x) = 1$, while the membership function for the linguistic value “does not fulfill” is $\mu_A(x) = 0$. The value of the membership function for the linguistic value “partially fulfills” and “partially does not fulfill” when trapezoidal and triangular membership functions are used, the function argument is calculated by coefficients k_l i k_r , respectively. Coefficients represent the percentage of what the evaluator describes as the particular parameter partially fulfilling of partially not fulfilling. The value of the membership function for the value “partially fulfills” and “partially does not fulfill” is calculated as follows:

- Calculation of arguments p_l and p_r of the membership function based on the coefficients k_l and k_r as follows:

- $p_l = a + \frac{(m-a)}{100} k_l$, for $k_l \leq 50\%$, whereas p_l is a numeric value of linguistic variable “partially fulfills”.
- $p_r = n + \frac{(b-n)}{100} k_r$, for $k_r \leq 50\%$, whereas p_r is a numeric value of linguistic variable “partially does not fulfill”.
- Calculation of the membership function value for p_l and p_r as follows:
 - $\frac{x-a}{m-a}$ whereas $x = p_l$ for $p_l \leq \frac{m-a}{2}$;
 - $\frac{b-x}{b-n}$ whereas $x = p_r$ for $p_r \leq \frac{m-a}{2}$.

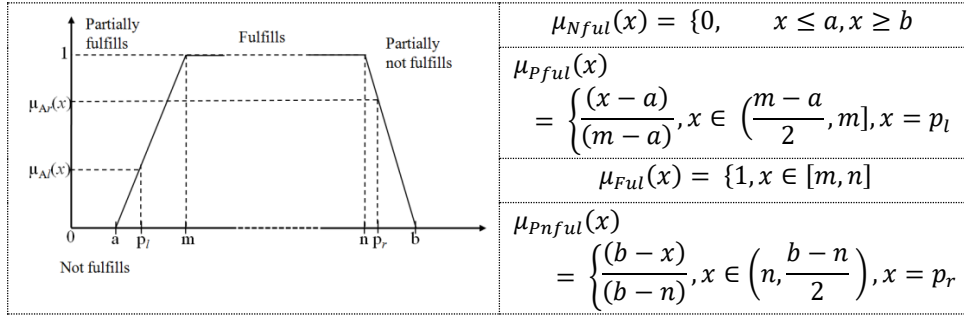


Fig. 4 Membership functions for linguistic values based on L- and R- trapezoid interval

The triangular membership function for the above linguistic values is applied in the same manner, with the value $m=n$. Figure 5. shows the triangular membership function for linguistic values.

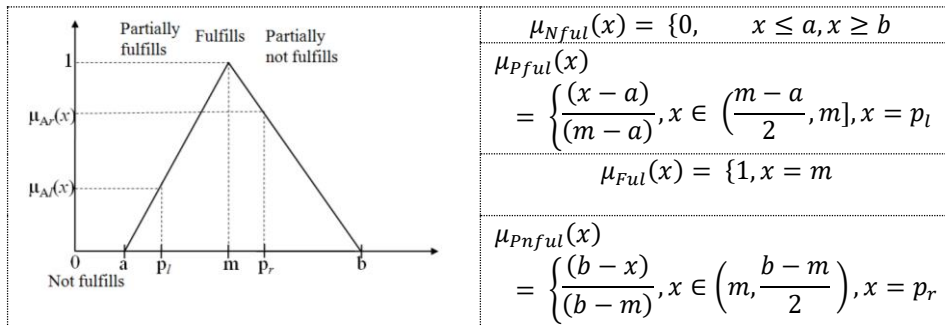


Fig. 5 Membership functions for linguistic values based on triangular function

4.5. Setting priorities and satisfaction degrees

Parameters for measuring the requirement quality in the given model do not have the same effect on decision-making. It is not sufficient to divide parameters into significant and less significant ones; instead, the degree of parameter effect on the end result must be defined. The degree is expressed in percentage. Therefore, priority is being determined for each of the parameters in the process of decision-making. Parameters with almost no effect on the end result are less significant with threshold limits that must not be crossed.

The limit that must not be crossed is the satisfaction threshold and is expressed in percentage, just as priorities.

Priorities and satisfaction threshold are given immediately after fuzzy values in square brackets [P_value, T_value], and relate to the fuzzy value itself. These values may be from the interval [0,1]. If the priority for a fuzzy value is not given, then its value is 1. In case there are no limitations at all, then the satisfaction threshold is not given as its assumed value is 0. If we present the priority and satisfaction threshold for the fuzzy linguistic variable as [P_0.7, T_0.4], this means that the priority of the variable is 70% while the satisfaction threshold is 40%.

The degree of constraint satisfaction is calculated by the use of the t-conorms S_p , whereas $S_p(\mu_{R_x^f}(\vartheta), 1 - \rho(R_x^f))$. ϑ is the observed value with membership degree $\mu_{R_x^f}(\vartheta)$ and priority $\rho(R_x^f)$.

4.6. Setting criteria and calculation of the general constraint satisfaction degree

Before calculating the general constraint satisfaction degree, it is necessary to define interdependence between constraint sets, or parameters. Constraint sets may be conjunctively and/or disjunctively related. An example of the equation for calculation of the general constraint satisfaction degree with conjunctive and disjunctive relations existing between three constraint sets is $(R_1^f \vee R_2^f) \wedge R_3^f$.

$$\alpha = T_L \left(S_L \left(S_p \left(\mu_{R_1^f}(\vartheta), 1 - \rho(R_1^f) \right), S_p \left(\mu_{R_2^f}(\vartheta), 1 - \rho(R_2^f) \right) \right), S_p \left(\mu_{R_3^f}(\vartheta), 1 - \rho(R_3^f) \right) \right) \quad (2)$$

General constraint satisfaction degree α in case of conjunctive dependence is calculated as t-norm T_L , while in the case of disjunctive dependence as S_L dual t-conorm of T_L .

The equation further goes:

$$S_p(x, y) = x + y - xy \quad (3)$$

$$T_L(x, y) = \max(x + y - xy, 0) \quad (4)$$

$$S_L(x, y) = \min(x + y, 1) \quad (5)$$

4.7. Decision-making

A decision on a requirement quality is made based on the calculated general constraint satisfaction degree and specific satisfaction degree for each variable. A minimal quality threshold is set. If the calculated general constraint satisfaction degree below the quality threshold, then the requirement should be re-defined. Otherwise, the requirement specification satisfies the required criteria.

If the general constraint satisfaction degree goes above the quality threshold, it is necessary to check if the satisfaction degree for each parameter is fulfilled in order to make the final decision. In case that any of the parameters do not fulfill the satisfaction degree, it is necessary to re-define the requirement so as for parameters to satisfy the required degree.

Upon requirement re-definition, evaluation of requirement is done entirely again for all, not only for the quality parameters that needed to be improved.

5. APPLICATION OF REQUIREMENT QUALITY MEASUREMENT MODEL

The model has been applied to determine the requirement quality for the design of PKI in the Ministry of Defense and Serbian Armed Forces. Upon application, the analysis of the requirement quality in all SDLC phases was performed. Two key reasons for a great number of incomplete requirements and abandoned requirements were identified. The first one is the lack of a requirement systematization for PKI to focus on the key ones, while the second reason was the inability to measure the requirement quality. The first issue was resolved by the development of the PKI classification scheme [8], while the second issue was resolved by designing the model for requirement quality measurement by application of the GPFCS system.

5.1. Selection of parameters for measuring the requirement quality

The analysis of requirements for the development of software for certification body proved that the major issue was requirements that needed to be re-examined and/or re-defined [8]. The analysis considered the most common characteristics that were not well defined in requirements but affected their quality.

Upon analysis of requirements that needed to be re-defined (or subsequently defined), the following lack of satisfaction of the requirement characteristics was identified:

Completeness. Completeness means that a requirement fully describes the functionality of the system to be implemented. The requirement must include all information needed for functionality design and implementation. The requirement must be complete before initiation of the product development life cycle it relates to.

Correctness. The requirement describes correctly the functionality to be implemented. Incorrectness in requirements makes deficiencies in architecture and implementation.

Feasibility. This characteristic is about the feasibility of implementation of the requirement within known constraints of the system and work environment. Requirements are of no use if the development team is not able to implement them. All relevant constraints must be considered to have individual requirements feasible.

Necessity/Usability. Implementation of requirements must create the value that users truly need or something they need for adjustments with requirements of the external system, interface, or standard.

Unambiguity. All users of the requirement specification should come up with a single, consistent interpretation. Requirements should be written in simple, concise, understandable, and clear language. Users of a requirement specification must be able to comprehend the requirement. It is necessary to define all specialized terms and ambiguous terms to create a better understanding of a requirement.

Verifiability. The requirement should be specified in a way that allows testing upon implementation to prove its fulfillment. Verification of requirements should be done via tests or some other ways, like inspection or demonstration, in order to determine if it was correctly implemented in the product.

5.2. Quality analysis of requirement selected parameters

The requirement quality analysis was performed for each of the above-mentioned parameters by answering previously defined questions. The purpose of the analysis was to collect data to define the membership function for constraints. The following questions were used in the requirement quality analysis:

1. Question for completeness:
 - Is the requirement described as fully as possible?
 - Are there any deficiencies in the requirement description related to functionality, performances, interface, environment, training, operability, safety, security?
 - Are all assumptions explicitly presented?
 - Does the requirement say what is needed, not how it is realized?
 - Is there a need to further elaborate or clarify the requirement?
 - Is each identified requirement a single requirement and not a set of requirements?
2. Questions for correctness:
 - Is the requirement described in a such way that required functionality can be implemented?
 - Does the requirement fulfill all or a part of real needs?
 - Is the requirement precise elaboration of a requirement of a higher level?
 - Are numeric values given in the requirement correct?
 - Is each linguistic requirement grammatically correct?
3. Questions for feasibility:
 - Can the requirement be implemented considering the existing hardware or software technologies?
 - Can each requirement be implemented considering the budget?
 - Can each requirement be implemented considering the project schedule?
 - Can each requirement be implemented considering the limitations of the staff (for example, enough staff, competencies, and experience)?
4. Questions for necessity/usability:
 - Is the requirement necessary for the success of the application or component?
 - Is the requirement mandatory?
 - Is the requirement mandatory to some the stakeholders, users, or organizations?
 - Does the requirement have any unnecessary limitations (for example, in architecture, design, implementation, testing, and other technology decisions)?
5. Questions for unambiguity:
 - Is each requirement clear and precise?
 - Is the meaning of each requirement objective and not subjective?
 - Is each requirement concise (meaning without any unnecessary and irrelevant information)?
 - Does each requirement have only one obvious interpretation?
 - Will interpretation of each requirement be the same for those who wrote it and those who interpret its specification?
 - Does each requirement use specific expressions?
6. Questions for verifiability:
 - Can the system be tested, demonstrated, reviewed, or analyzed upon implementation to prove the requirement is fulfilled?
 - Can verification criteria be identified?
 - Are requirements stated precisely to make criteria and requirement specification for successful testing easier?
 - Is requirement specification created in a such way as not to use unverifiable terms like flexible, light, sufficient, safe, proper, useful, usable, as needed, suitable, easy, small, big, maximal, minimal?

Three independent evaluators evaluated the quality of 30 requirements rejected as bad ones. The evaluation was performed by answering the questions for each of the parameters.

The allowed answers were: “does not fulfill”, “partially does not fulfill”, “partially fulfills” and “fulfills”. Upon evaluation completion, normalization of results was done as shown in Table 1 below.

Table 1 Quality measurement of rejected requirements

Parameter	Values			
	Does not fulfill	Partially fulfills	Fulfills	Partially does not fulfills
Completeness	9	22	18	41
Correctness	28	27	13	22
Feasibility	23	32	9	26
Usability	12	18	22	38
Unambiguity	27	18	17	28
Verifiability	18	32	12	28

5.3. Design of membership function for fuzzy constraints

Upon equality measurement of rejected requirements and obtained results, membership functions for constraints are designed. It is necessary to define variables and values to apply fuzzy logic. Linguistic variables are parameters while linguistic values are answers on the quality of requirements.

Linguistic variables and linguistic values are defined as follows:

- Variable X_1 is linguistic variable *Complete* with the domain d_1 ;
- Variable X_2 is linguistic variable *Correct* with the domain d_2 ;
- Variable X_3 is linguistic variable *Feasible* with the domain d_3 ;
- Variable X_4 is linguistic variable *Usability* with the domain d_4 ;
- Variable X_5 is linguistic variable *Unambiguous* with the domain d_5 ;
- Variable X_6 is linguistic variable *Verifiable* with the domain d_6 .

Linguistic values for all linguistic variables are: *Does not fulfill*, *Partially fulfills*, *Fulfills*, *Partially does not fulfills*

The constraint set is identified: $R_1^f = \text{"Optimal completeness"}$, $R_2^f = \text{"Optimal correctness"}$, $R_3^f = \text{"Optimal feasibility"}$, $R_4^f = \text{"Optimal usability"}$, $R_5^f = \text{"Optimal ambiguity"}$, $R_6^f = \text{"Optimal verifiability"}$. Given constraints may be defined as linguistic fuzzy variables, while their membership functions for linguistic values respond to the functions presented in Fig. 6 and Fig. 7.

5.4. Definition of satisfaction and constraint thresholds

Upon definition of the membership functions, priorities and satisfaction thresholds for linguistic variables were identified:

- "Optimal completeness", [P_0.8, T_0.92], i.e. $[\rho(R_1^f) = 0.8, T_0.92]$;
- "Optimal correctness", [P_0.85, T_0.92], i.e. $[\rho(R_2^f) = 0.85, T_0.92]$;
- "Optimal feasibility", [P_0.9, T_0.92], i.e. $[\rho(R_3^f) = 0.9, T_0.92]$;
- "Optimal usability", [P_0.9, T_0.92], i.e. $[\rho(R_4^f) = 0.9, T_0.92]$;
- "Optimal ambiguity", [P_0.85, T_0.92], i.e. $[\rho(R_5^f) = 0.85, T_0.92]$;
- "Optimal verifiability", [P_0.9, T_0.92], i.e. $[\rho(R_6^f) = 0.9, T_0.92]$.

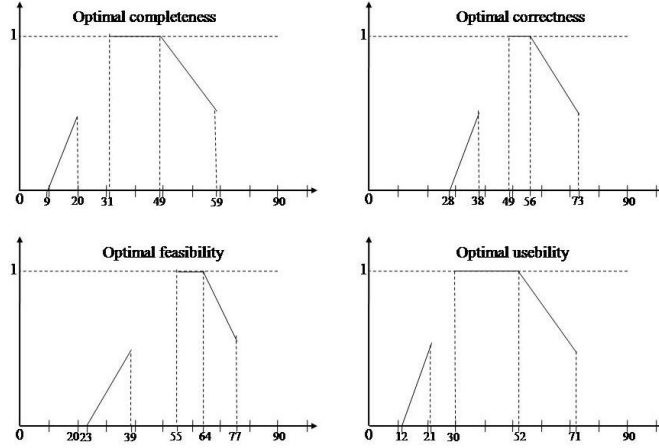


Fig. 6 Membership functions for: "Optimal completeness", "Optimal correctness", "Optimal feasibility", "Optimal usability"

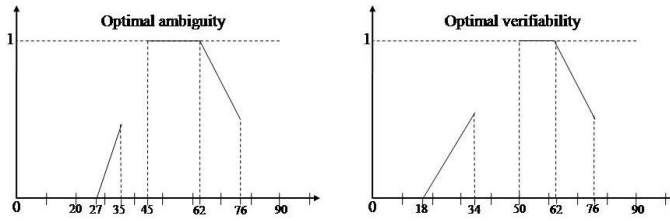


Fig. 7 Membership functions for: "Optimal ambiguity" and "Optimal verifiability"

Before calculation of the general degree of constraint satisfaction, it is necessary to identify dependence between constraint sets, i.e. parameters. Constraint sets may be conjunctively and/or disjunctively related.

The quality of requirement depends on the fulfillment of constraint conditions of all variables; therefore, the conjunctive dependence relation between constraint sets was selected.

5.5. Calculation of the general constraint satisfaction degree and decision-making

The equation for calculation of the general constraint satisfaction degree for six conjunct constraint sets, $R_1^f \wedge R_2^f \wedge R_3^f \wedge R_4^f \wedge R_5^f \wedge R_6^f$, is complex. For easier calculation, and owing to the associativity, the constraint sets will be conjunct as follows: $(R_1^f \wedge R_2^f) \wedge R_3^f \wedge (R_4^f \wedge R_5^f) \wedge R_6^f$. Now, the equation for calculation of the general quality constraint satisfaction degree of a requirement is:

$\alpha = T_L(\alpha_1, \alpha_2)$, whereas:

$$\alpha_1 = T_L \left(T_L \left(S_P \left(\mu_{R_1^f}(\vartheta), 1 - \rho(R_1^f) \right), S_P \left(\mu_{R_2^f}(\vartheta), 1 - \rho(R_2^f) \right) \right), S_P \left(\mu_{R_3^f}(\vartheta), 1 - \rho(R_3^f) \right) \right)$$

$$\alpha_2 = T_L \left(T_L \left(S_P \left(\mu_{R_4^f}(\vartheta), 1 - \rho(R_4^f) \right), S_P \left(\mu_{R_5^f}(\vartheta), 1 - \rho(R_5^f) \right) \right), S_P \left(\mu_{R_6^f}(\vartheta), 1 - \rho(R_6^f) \right) \right)$$

For every ϑ value, its membership degree $\mu_{R_x^f}(\vartheta)$, obtained as a cross-section of membership function and current value is calculated, as described in Chapter 4.4. Values for this example are given in Tables 2 and 3.

Table 2 Calculation of the membership degree for complete, correct, feasible

Examples	Requirement specification					
	X_1 - Complete	$\mu_{R_1^f}(\vartheta)$	X_2 - Correct	$\mu_{R_2^f}(\vartheta)$	X_3 - Feasible	$\mu_{R_3^f}(\vartheta)$
a)	partially does not fulfills – 15	0.85	partially does not fulfills – 10	0.90	fulfills	1
b)	partially does not fulfills – 5	0.95	partially does not fulfills – 5	0.95	fulfills	1
c)	partially fulfills – 50	0.5	fulfills	1	fulfills	1

Table 3 Calculation of the membership degree for usability, unambiguous, verifiable

Examples	Requirement specification					
	X_4 - Usability	$\mu_{R_4^f}(\vartheta)$	X_5 - Unambiguous	$\mu_{R_5^f}(\vartheta)$	X_6 - Verifiable	$\mu_{R_6^f}(\vartheta)$
a)	fulfills	1	partially does not fulfills – 8	0.92	partially does not fulfills – 13	0.87
b)	fulfills	1	partially does not fulfills – 8	0.92	partially does not fulfills – 10	0.90
c)	fulfills	1	fulfills	1	fulfills	1

If values for calculated membership degree of every $\mu_{R_x^f}(\vartheta)$ and priority values $\rho(R_x^f)$ put together in the equation for calculation of the general constraint satisfaction degree α , the following results are obtained.

$$\begin{aligned} \alpha_{1a} &= T_L \left(T_L \left(S_P(0.85, 1 - 0.8), S_P(0.90, 1 - 0.85) \right), S_P(1, 1 - 0.9) \right) \\ \alpha_{1a} &= T_L(T_L(0.88, 0.915), 1) = T_L(0.795, 1) = 0.795 \\ \alpha_{2a} &= T_L \left(T_L \left(S_P(1, 1 - 0.9), S_P(0.92, 1 - 0.85) \right), S_P(0.87, 1 - 0.9) \right) \\ \alpha_{2a} &= T_L(T_L(1, 0.932), 0.883) = T_L(0.932, 0.883) = 0.815 \\ \alpha_a &= T_L(0.795, 0.815) = 0.61 \end{aligned}$$

Testing of the model led to identifying the following conditions for measurement the requirement quality:

- If the global constraint satisfaction degree is larger than 0.7, $\alpha > 0.7$, than the requirement specification is deemed of good enough quality for implementation;
- The satisfaction degree threshold for any variable must not be less than 0.92, meaning that the satisfaction degree must be higher than 92%.

Based on the above-stated conditions for measuring the requirement quality and calculated global constraint satisfaction degree (0.61), it is obvious that the requirement specification does not fulfill the required quality. Besides, parameters “Complete”, “Correct”, and “Verifiable”, with satisfaction degrees of 0.88, 0.915, and 0.883 respectively, do not fulfill the requirement of requirement degree being over 0.92. This means that it is necessary to put a special emphasis on improvements of these parameters in the requirement re-definition process.

Upon re-definition of the requirement specification, the quality of the above-mentioned parameters was improved. A new evaluation of the quality was done by calculating the general constraint satisfaction degree.

$$\begin{aligned}\alpha_{1b} &= T_L \left(T_L (S_P(0.95, 1 - 0.8), S_P(0.95, 1 - 0.85)), S_P(1, 1 - 0.9) \right) \\ \alpha_{1b} &= T_L (T_L(0.96, 0.9575), 1) = T_L(0.9175, 1) = 0.9175 \\ \alpha_{2b} &= T_L \left(T_L (S_P(1, 1 - 0.9), S_P(0.92, 1 - 0.85)), S_P(0.9, 1 - 0.9) \right) \\ \alpha_{2b} &= T_L (T_L(1, 0.932), 0.91) = T_L(0.932, 0.91) = 0.842 \\ \alpha_b &= T_L(0.9175, 0.842) = 0.7595\end{aligned}$$

The calculated general constraint satisfaction degree is higher than the minimal constrain of the good quality requirement. However, the satisfaction threshold for the parameter “Verifiable” is 0.91 and that is lower than the satisfaction threshold. Due to this, non-fulfillment of this parameter, the requirement specification is deemed insufficiently good enough; therefore, it is necessary to improve the specification of the verifiability characteristic.

The model is designed in a way that if only one parameter’s value is “partially fulfills” with a minimal percentage, while all other parameters’ values are “fulfills”, the specification will be rejected as being of not good enough quality.

$$\begin{aligned}\alpha_{1c} &= T_L \left(T_L (S_P(0.5, 1 - 0.8), S_P(1, 1 - 0.85)), S_P(1, 1 - 0.9) \right) \\ \alpha_{1c} &= T_L (T_L(0.6, 1), 1) = T_L(0.6, 1) = 0.6 \\ \alpha_{2c} &= T_L \left(T_L (S_P(1, 1 - 0.9), S_P(1, 1 - 0.85)), S_P(1, 1 - 0.9) \right) \\ \alpha_{2c} &= T_L (T_L(1, 1), 1) = T_L(1, 1) = 1 \\ \alpha_c &= T_L(0.6, 1) = 0.6\end{aligned}$$

The calculation proves that the specification does not meet the set quality standard as the calculated general constraint satisfaction degree is 0.6 which is lower than the quality threshold of 0.7. Besides, the satisfaction threshold for the parameter “Complete” is 0.6 which is lower than the set satisfaction threshold of a minimum of 0.92.

6. CONCLUSION

The paper focuses on measuring the requirement quality based on the characteristics that define good requirements. The requirement quality is measured by the application of the Generalized Prioritized Fuzzy Constraint Satisfaction Problem. Application of GFPCPS enables identification of the share (priority) of participation of the quality of the requirement characteristics in the calculation of the general satisfaction degree. Based on the calculated general satisfaction degree, and based on the defined constraint and satisfaction degree thresholds of specific characteristics, a decision on the fulfillment of the requirement quality is made. The model may be used by experts and stakeholders for evaluation of the quality of well-defined requirements, as well as for requirement quality validation before the official approval of the requirement. Practical application of the model, together with the PKI requirement classification scheme, proved good results in reducing the number of incomplete and poorly defined requirements. The model may be used also for the evaluation of the requirement quality at all lifecycle stages of the software product. The model is a flexible one as it enables the selection of different requirement characteristics and designing of the membership functions of constraints, as per the experience of experts.

REFERENCES

- [1] The Standish Group. Chaos manifesto: Think big, act small. 2013
- [2] K. Pohl, Requirements engineering: Fundamentals, principles, and techniques. Berlin, Germany: Springer-Verlag, 2010.
- [3] N. Mulla and S. Girase, "A new approach to requirement elicitation based on stakeholder recommendation and collaborative filtering", *International Journal of Software Engineering and Applications (IJSEA)*, 3(3), 51-60, 2012.
- [4] L.R. Wong, and D.S. Mauricio, "New factors that affect the activities of the requirements elicitation process", *Journal of Engineering Science and Technology (JESTEC)*, 13(7), 1992-2015, 2018.
- [5] B. Davey and K. Parker, "Requirements elicitation problems: A literature analysis Issues in informing", *Science and Information Technology*, 12, 71-82, 2015.
- [6] I. Hadar, P. Soffer and K. Kenzi, "The role of domain knowledge in requirements elicitation via interviews: An exploratory study", *Requirements Engineering*, 19(2), 143-159, 2014.
- [7] D.C. Derrick, A. Read, C. Nguyen, A. Callens and G.-J. de Vreede, "Automated group facilitation for gathering wide audience end-user requirements". *Proceedings of 46th Hawaii International Conference on System Sciences*. Wailea, Maui, Hawaii, pp. 195-204, 2013.
- [8] R. Prodanović and I. Vulić, "Classification as an approach to public key infrastructure requirements analysis", *IET Software*, vol. 13, no. 6, pp. 518-527, 2019, doi: 10.1049/iet-sen.2018.5286.
- [9] Ludwig Consulting Services, LLC. Managing requirements. 2009, Retrieved May 7, 2018, from www.jiludwig.com.
- [10] G. Génova, J. M. Fuentes, J. Llorens, O. Hurtado, V. Moreno, "A Framework to Measure and Improve the Quality of Textual Requirements", *Requirements Engineering* 18(1), 2013, DOI: 10.1007/s00766-011-0134-z
- [11] B. Heinrich, D. Hristova, M. Klier, A. Schiller, M. Szubartowicz, "Requirements for Data Quality Metrics", *Journal of Data and Information Quality*, Volume 9, Issue 2, January 2018, <https://doi.org/10.1145/3148238>
- [12] R. J. Halligan, "Requirements Quality Metrics: The Basis of Informed Requirements Engineering Management", Presented at the 1993 Complex Systems Engineering Synthesis and Assessment Technology Workshop (CESAW '93), Calvados, MD, USA
- [13] P. S. Kummier, L. Vernisse, H. Fromm, "How Good are My Requirements?: A New Perspective on the Quality Measurement of Textual Requirements", 11th International Conference on the Quality of Information and Communications Technology (QUATIC), Coimbra, Portugal, 4-7 Sept. 2018, DOI: 10.1109/QUATIC.2018.00031
- [14] L. R. Wong and D. S. Mauricio, "Qualities that the Activities of the Elicitation Process Must Meet to Obtain a Good Requirement", *Journal of Engineering Science and Technology* Vol. 14, No. 5, pp. 2883 – 2912, 2019.
- [15] B. Boehm, "A view of 20th and 21st century software engineering". ICSE '06 Proceedings of the 28th international conference on Software engineering. University of Southern California, University Park Campus, Los Angeles, CA: Association for Computing Machinery, ACM New York, NY, USA. pp. 12–29. 2006. ISBN 1-59593-375-1.
- [16] M.A.Wick, E. Iem, M.A.J Burns, et al.: 'A guide to the business analysis body of knowledge® (BABOK® Guide)', v3, IIBA, 2015, p. 16
- [17] IEEE SA - 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology.
- [18] A. M. Davis, *Software Requirements: Objects, Functions, and States*, Second Edition. Prentice Hall. 1993, ISBN 978-0-13-805763-3.
- [19] IEEE Computer Society, *IEEE Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers, Inc. 1998, ISBN 978-0-7381-0332-7.
- [20] G. Koelsch, "Requirements Writing for System Engineering", Springer Science+Business Media New York, New York, 2016, DOI 10.1007/978-1-4842-2099-3
- [21] D. Firesmith, "Specifying Good Requirements", *Journal of Object Technology*, Vol. 2, No. 4, 2003
- [22] X. Luo, J. Lee, H. Leung and N.Jennings, "Prioritized Fuzzy Constraint Satisfaction Problems: Axioms, Instantiation and Validation". *Fuzzy Sets and Systems*, 136(2), pp. 151–188, 2003
- [23] G. Panić, *The development a dedicated system for the application fuzzy logic in systems for manage of XML documents*, Doctoral thesis, Faculty of Sciences, University of Novi Sad, 2013
- [24] A. Takači, *Trougaone norme prioriteta i njihova primena na modeliranje ispunjenja fazi ograničenja*, Doctoral thesis, Faculty of Sciences, University of Novi Sad, 2006
- [25] NASA, Appendix C: How to Write a Good Requirement, 2019, <https://www.nasa.gov/seh/appendix-c-how-to-write-a-good-requirement>.